

Durée : 1h30 – Tous documents autorisés

■ ■ ■ Utilisation de LEX — 6 points

1– L'inventeur d'un **jeu de carte de stratégie** vous contacte pour réaliser une version « *online* » de son jeu.

6pts Pour commencer, il vous présente la version simplifiée suivante :

- le jeu consiste en un paquet de 60 cartes, chacune pouvant être « *offensive* », « *défensive* » ou les deux ;
- sur chaque carte il est noté deux caractéristiques « Protection » et « Dommages » suivies d'une valeur de bonus, coefficient positif, ou de malus, coefficient négatif ;
- deux joueurs, « Joueur1 » et « Joueur2 », s'affrontent :
 - ◇ chaque joueur tire 3 cartes du paquet ;
 - ◇ il en sélectionne 2 parmi les 3 et les pose au centre de la table ;
 - ◇ on détermine qui remporte le « *tour* » de jeu :
 - * chaque joueur fait la somme des points de Dommages, « D », et des points de Protection « P », des cartes qu'il a déposé :
 - ▷ Joueur1 dispose de D_1 et P_1
 - ▷ Joueur2 dispose de D_2 et P_2
 - * pour « Joueur1 », on calcul $R_1 = P_1 - D_2$;
 - * pour « Joueur2 », on calcul $R_2 = P_2 - D_1$;
 - ◇ le joueur qui possède la valeur de R la plus élevée, c-à-d qui a subi le moins de dommages, a gagné le tour.
- il y a 10 tours (6 cartes tirées par tour sur 60) : celui qui remporte le plus de tour a gagné.

Le fichier qui contient le déroulement d'un tour de jeu est le suivant :

```
Joueur1: +3D, 0P;
Joueur2: +5D, -1P;
Joueur1: 0D, +4P;
Joueur2: +1D, +1P;
```

▷ $R_1 = (0+4)P - (+5+1)D = 4-6 = -2$
 ▷ $R_2 = (-1+1)P - (+3+0)D = 0-3 = -3$
 C'est Joueur1 qui l'emporte !

Remarque : les lignes correspondant à la pose d'une carte par un joueur sont données dans le fichier dans n'importe quel ordre.

Autre exemple :

```
Joueur1: +2D, 0P;
Joueur1: +1D, +1P;
Joueur2: +2D, +1P;
Joueur2: +4D, 0P;
```

▷ $R_1 = (0+1)P - (+2+4)D = 1-6 = -5$
 ▷ $R_2 = (+1+0)P - (+2+1)D = 1-3 = -2$
 C'est Joueur2 qui l'emporte !

Questions :

- a. Écrire le programme LEX réalisant l'analyse du fichier : (5pts)
 - vérifie que chaque joueur a posé deux cartes pour un tour et si ce n'est pas le cas, indique quel est le joueur qui a mis trop de cartes ;
 - affiche si Joueur1 ou Joueur2 remporte le tour ;

Pour cette question, on considérera que le fichier contient toujours 4 lignes.
- b. Si le fichier contient moins de 4 lignes comment modifier votre analyseur pour prévenir l'utilisateur et arrêter l'analyse ? (1pt)

■ ■ ■ LEX & YACC — 6 points

2– Écrire le l'analyseur YACC permettant de gérer une partie :

- 6pts** ▷ vérifier qu'il y a bien 10 tours de jeu ;
 ▷ désigner le vainqueur de la partie, c-à-d celui qui a remporté le plus grand nombre de tours.

■ ■ ■ XML, DTD & XSLT — 8 points

3– L’inventeur veut passer maintenant à une phase de commercialisation où chaque carte va :

- 8pts ▷ être décorée avec une illustration faite par un artiste ;
▷ contenir une description de l’objet, de l’événement ou de l’entité associé à la carte ;
▷ disposer d’un nom ;
▷ avoir son nombre d’exemplaire dans le paquet de jeu.

Le but étant d’amener les joueurs à les collectionner en fonction de leur rareté (l’inventeur imagine faire des paquets contenant des cartes différentes et pouvoir les vendre).

Une **carte** est décrite par :

- * son nom ;
- * sa description ;
- * son illustration donnée par une URL ;
- * ses points de dommage ;
- * ses points de protection ;
- * son nombre dans le paquet exprimé par : unique, double ou triple.

Questions :

- a. Proposez un DTD permettant de décrire toutes les cartes. (3pts)
- b. Écrire un fichier au format XSLT pour chacun des affichages suivant au format HTML : (4pts)
 - a. le nom de la carte ;
 - b. ses points de dommage et de protection ;
 - c. son illustration.
- c. Donnez la requête XPath permettant de compter le nombre de cartes uniques. (1pt)